REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

FINAL/01 AUG 88 TO 31 JUL 92

COMMPUTATIONAL METHODS FOR FLOW PROBLEMS - PARALLEL ALGORITHMS, FLOW CONTROL, AND NOVEL APPROACHES

2305/BS
AFOSR-88-0274
61102F

PROFESSOR GAUDIOT

UNIVERSITY OF CALFIRONIA
DEPARTMENT OF COMPUTER SCIENCE
AND SECTION FOR NEUROBIOLOGY
LOS ANGELES CA 90089-1147

AFOSR-TR- 95 - 0291

AFOSR/NM
110 DUNCAN AVE, SUTE B115
BOLLING AFB DC 20332-0001

AFOSR-88-0274

DTIC ELECTE JUN 19 1995 F

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

We have created an object recognition system, in the context of the general goal of contributing to the development of a visual architecture. The system makes use of wavelet transforms, of dynamic link matching, and is of general neural style. We have implemented the system in several versions, as an object-oriented modular program on a workstation, and as a parallel farm structure on an array of transputers. Object recognition from camera images is invariant to translation, scaling and rotation in the image plane. and is robust with respect to lighting and to rotation in depth. We have tested the system on the task of recognizing human faces. With galleries of about 90 faces, the system achieved highly confident recognition on ca. 85% of the input images.

DTIC QUALITY INSPECTED 5

19950615 026

OF REPORT	OF THIS PAGE	OF ABSTRACT	
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR(SAME AS REPORT)

Final Report, Project AFOSR 88-0274

Christoph von der Malsburg, Principal Investigator
Department of Computer Science and Section for Neurobiology
University of Southern California

Abstract

We have created an object recognition system, in the context of the general goal of contributing to the development of a visual architecture. The system makes use of wavelet transforms, of dynamic link matching, and is of general neural style. We have implemented the system in several versions, as an object-oriented modular program on a workstation, and as a parallel farm structure on an array of transputers. Object recognition from camera images is invariant to translation, scaling and rotation in the image plane, and is robust with respect to lighting and to rotation in depth. We have tested the system on the task of recognizing human faces. With galleries of about 90 faces, the system achieved highly confident recognition on ca. 85% of the input images.

1 Introduction

1.1 General Goals of the Project

This project was pursued by the PI (C. v.d. Malsburg) as part of a longer-term effort to contribute to the development of a cognitive architecture, or slightly more specifically, a visual architecture, to be formulated in neural style. This effort is shaped by the conviction that significant progress in that direction can only be achieved by seriously attempting to solve specific problems, as for instance recognition and representation of objects in real visual scenes, but was pursued with the more general goal in mind.

Based on the belief that the brain will eventually by understood in terms of a relatively simple conceptual system, called cognitive architecture, this project attempted to formulate better constraints for the construction of that system. One of the necessary constraints seems to be neural style. By that we here understand two aspects. One, formulation of a system that is amenable to fine-grained parallelism. Two, emphasis on principled and generic approaches, in contrast to a style in which each new problem is solved with the help of its own new algorithm, complete with its own special data structure—which used to be the original approach of Artificial Intelligence. Rather, neural style calls for a few basic organizational principles and data structures that can be specialized (as much as possible by automatic processes) to particular situations and problems. It has been one of our design criteria to willingly pay the price for simplicity and genericness in terms of data volume and processing complexity. On the side of technology this strategy will pay off since with falling costs of data storage and computing power and the rising cost and complexity of software, emphasis must be

placed on simplicity of design. The eventual goal of neural systems must be the replacement of a continued design effort by once-for-all design complemented by methods of autonomous adaptation.

A word of caution is in order here. There is the wide-spread belief that present-day neural systems, based on McCulloch-Pitts type neurons and adaptive methods like back-propagation, will be all that is needed to realize the neural style. Unfortunately it has turned out that it is not possible to just start on any problem with a standard network and train it automatically by exposure to a statistical ensemble of example cases, the problem being that in going from small to realistically large problems the number of examples required grows to an unrealistic, astronomical level—if there is convergence at all. The standard solution to this scaling problem is to start the system with a network structure already specialized to a large extent to the problem at hand. Unfortunately, this just brings us back to the original problem—special system design for special problems. In contrast, natural brains have tremendous power to generalize from very few examples. This seems to indicate that there are fairly general and yet at the same time powerful *a priori* structures. These seem to fit the natural environment just like a glove fits a hand.

In previous work [21, 17] we had come to the conclusion that many of the problems with neural networks can be solved if another layer of variables is added, over and above the level of neural signals—variables that we came to call "dynamic links" [22]. These variables serve approximately the same purpose as pointers in classical data structures, especially, binding more elementary symbols to each other to form complex symbols. In the project reported here it was our main goal to develop dynamic links further, and while applying them to the specific problem of invariant object recognition we made efforts to solve some of the fundamental problems with neural networks.

1.2 Special Goal of the Project: Invariant Object Recognition

We set out to emulate as far as as possible the ability of our visual system to recognize and represent objects and to remember new objects from a single exposure. Object recognition is a demanding problem since a given object may appear in an infinity of ways to the eye or camera, due to changes in position, size and orientation, rotation in depth, changes in background and lighting, and due to deformations of the object itself. Moreover, our visual system is able to remember new objects from a single exposure, without lengthy statistical learning from many exemplars. We therefore implemented the basic capability of object recognition without the requirement of statistical learning (although we realize that statistical parameter estimation is required for the performance of certain discrimination tasks).

Although the concrete numerical measure of success for our system is its ability to classify objects, in its construction we insisted on object representation as well. Although logically this may not be required for classification, flexible handling of objects can only be realized on the basis of an explicit object representation.

In the spirit of our general goals we kept ourselves from any conscious attempt at specializing our system to particular object classes. Although in most of our experiments and evaluations we let the system recognize human faces, we also showed its ability to discriminate arbitrary objects [11].

During all of the development of the system we had in mind creating a basis for later extensions to a more complete visual architecture and the potential for the implementation of further capabilities, such as derivation of object properties (especially surface shape description) from image features, decomposition into object parts, and object tracking and handling.

1.3 The Four-Layer Perceptron and the Dynamic Link Architecture

Implicitly, Rosenblatt's 4-layer perceptron is providing the perceptual background for much of present work on neural object recognition. It therefore is a natural point of departure for an exposition of the system we developed in this project. The four layers are called S, the sensory layer, in the form of a pixel array; $A^{(1)}$, the layer of position-dependent feature detectors ("associators"); $A^{(2)}$, the layer of position-independent feature detectors; and R, the layer of classification ("recognition") cells. Layer $A^{(1)}$contains one cell (α, x) for each feature type α and each position x. Layer $A^{(2)}$contains only one cell (α) for each feature type α. If feature α appears once or several times in S, the "invariant feature cell" (α) in $A^{(2)}$is fired, independent of the position(s) of the α-cell(s) in S. This is made possible by a large fan-in of connections from all the cells (α, x) for different x in $A^{(1)}$to the one cell (α) in $A^{(2)}$. Thus, the appearance of a pattern in S triggers the activity of a list of invariant feature detectors in $A^{(2)}$. With appropriate connections from $A^{(2)}$to R, a cell in that latter layer can be made to fire selectively to a type of pattern, irrespective of where it appeared in S.

The great weakness of the 4-layer perceptron (and Rosenblatt was acutely aware of this) is the fact that in the transition from $A^{(1)}$to $A^{(2)}$the system, while shedding the information it wants to shed—the position of the pattern in S—, involuntarily also has to discard all information on spatial relationships between features. This leads to the very concrete danger of ambiguity: there may be different patterns that are described in $A^{(2)}$by the same comprehensive list of features but that contain the features in a different spatial arrangement. Another difficulty is the lack of figure-ground discrimination in the 4-layer perceptron: the simultaneous presence of several objects in the scene in S leads to "illusory conjunctions", combinations of features which really belong to different objects in the scene but which simulate objects, not actually present, uniting just those features.

In the Dynamic Link approach these difficulties are solved in a simple and yet principled way: During presentation of an image in S, the signals sent from $A^{(1)}$to $A^{(2)}$are not constant but are made to fluctuate in time. Moreover, they are correlated with each other, and the correlations are shaped such as to express the spatial neighborhood relationships in $A^{(1)}$and in S. Strong correlations stand for small distance, weak correlations for large distance or for features that belong to different objects. Signal fluctuations are created by short-range excitatory connections in the layer $A^{(1)}$. In this way, the image of an object is represented in $A^{(1)}$by a topologically structured network of local feature cells and connections between neighbors among them, the connections being expressed in the signals to $A^{(2)}$in terms of signal correlations.

These correlations are to be decoded in layer $A^{(2)}$. This can be done with the help of object-specific networks in $A^{(2)}$. In order to store an object, a copy of the network representing it in $A^{(1)}$is created in $A^{(2)}$. Thus, $A^{(2)}$is a possibly large collection of model networks. When an object is shown, the system has to selectively activate the

one stored model network in $A^{(2)}$that corresponds to its structure. This is possible on the basis of a process of self-organization, called "dynamic link matching". It employs rapid synaptic switching of the connections between $A^{(1)}$and $A^{(2)}$and the connections within $A^{(2)}$. The result of this process is a selective activation of one of the stored model circuits in $A^{(2)}$as well as the connections between corresponding points inside the object in $A^{(1)}$and in $A^{(2)}$. The process makes use of a resonance phenomenon in which neighboring cells in $A^{(1)}$can easily fire neighboring cells in $A^{(2)}$. It is essential here that by our construction of connections in and between $A^{(1)}$and $A^{(2)}$we have created an overlapping system of densely coupled sets of cells, each set involving local clusters of cells in $A^{(1)}$and $A^{(2)}$and many connections between them. According to our construction, such double-clusters can only be formed if the model network in $A^{(2)}$contains the same local features as the active network in $A^{(1)}$*in the same arrangement*! The clusters of densely coupled cells now dominate the activity process, creating activity events ("active clusters") that involve pairs of corresponding locations in the object in $A^{(1)}$and its model in $A^{(2)}$. With the help of a kind of rapid and reversible Hebbian plasticity ("dynamic links"), connections between corresponding points in $A^{(1)}$and $A^{(2)}$and all the connections in the correct model network are selectively activated. This resonance is not possible with another model network, even if it contains the same summary list of features in another arrangement, since neighboring cells in $A^{(1)}$cannot be consistently connected to neighboring cells in $A^{(2)}$by feature-type preserving connections. The system thus solves the ambiguity problem of the 4-layer perceptron.

The price paid by the dynamic link architecture for these functional enhancements is the necessity of a period of rapid self-organization in response to the presentation of an image, before the system is ready for its reponse. We here explain only that minimum of the dynamic link architecture that is necessary to understand and motivate the work covered by this report. A more complete account is given in [21, 22, 17, 11].

2 Description of the Object Recognition System

The project reported here was a battle on two fronts: that of developing a feature set to encode images, and that of formulating the matching dynamics that goes on between the layers $A^{(1)}$(here called "image domain") and $A^{(2)}$(here called "model domain"). For each we have developed a successful strategy, which we will outline in this section. The bulk of our effort was spent on the development and testing of alternative forms for these two strategies. The result is distinguished by extreme simplicity and efficiency.

2.1 Image Preprocessing

In the above parlance, this stage extracts the features from the image presented in layer S and forms the representation in the image domain (layer $A^{(1)}$). The goal of the image coding stage is to represent the image in a data format that is robust with respect to variations in the image, that is complete (in the sense that the original image could be reconstructed from it), and that forms a natural basis for the ensuing process of matching against stored object models.

The first stage of the feature extraction process is based on convolutions. Connected to each feature type there is a kernel, acting as the receptive field or neural sensitivity

function for that feature type. The convolution kernels that we use typically have the form

$$\psi_{\vec{k}}(\vec{x_o}) = \frac{\vec{k}^2}{\sigma^2} \exp\left(-\frac{\vec{k}^2 \vec{x_o}^2}{2\sigma^2}\right) \left[\exp\left(i\vec{k}\vec{x_o}\right) - \exp\left(-\sigma^2/2\right)\right] ,$$

that is, they consist of a plane wave (with wave number vector $\vec{k}$, which determines the spatial frequency and orientation of the wave) that is restricted to a Gaussian window centered on position $\vec{x_o}$ of width $\sigma/|k|$, adjusted by the second term to be DC free. All of our experiments were based entirely on the wavelet concept. That is, we worked with families of similar kernels, each individual kernel being a rotated or scaled version of a prototypical kernel. Some effort went into a strategy for sampling the image appropriately, that is, into selecting a discrete set of $\vec{k}$ values and of image positions (details are described in [11]). Important selection criteria are the amount of detail retained from the image, the robustness of individual features against image variation, and robustness and efficiency of the matching process.

For the sake of the latter we let the linear wavelet transformation step be followed by a non-linear step. Each wavelet component of specific $\vec{k}$ and $\vec{x_o}$ contains a cosine and a sine part (corresponding to the real and the imaginary part of the above kernel). We squared the two parts and added them up (corresponding to the squared magnitude of the complex valued wavelet component). Although it is to be expected that the above goal of completeness of representation is thereby compromised, the resulting matching operation became much more reliable, efficient and robust.

The present version of our system is based on 4–6 resolution levels (spatial frequencies) of 8 orientations each. Spatial samples are arranged as "jets," each jet comprising a full family of wavelet types, all centered on the same $\vec{x_o}$. To represent an object we typically use 70 jets of 40 components each.

Considerable effort was invested over the life-time of the project into investigating improvements and alternatives to the data structures and matching procedures. One set of experiments centered on orthogonal wavelets, as developed by Y. Meyer and S. Mallat. These have the great advantages of permitting a fast transformation algorithm (with the same complexity as the Fast Fourier Transform) and precise and efficient image reconstruction, and of being optimal in the image coding sense. According to our experience, however, orthogonal wavelets are not sufficiently robust against image variation (as, for instance, small shifts relative to the sampling grid, or small rotations). We made repeated and intense attempts at the re-introduction of the phase (cos/sin ratio) abolished by taking jet magnitudes, to improve the precision of matching. The difficulty lies in the rapid variations of phases over the image, which leads to many local optima when comparing jets of a model to jets in the image. We have made progress on this issue, but have not been able to solve it yet.

Another excursion had (and has) to do with image-determined feature locations (as distinct from rigid sampling grids). We are particularly interested in resolution hierarchies of edge detectors. The advantage of this would be a reduction in the image data variability that is trivially introduced by the arbitrary positioning of sampling grids. This work could not be completed during the project, mainly because an irregular sampling grid necessitates a new matching algorithm.

2.2 The Matching Algorithm

Various practical considerations have made us diverge with the matching algorithm from the direct neural version suggested in the dynamic link architecture. The result is an efficient and fast elastic matching procedure which, however, is limited in its potential for further extensions. Much of the effort in this project was spent on the development of alternatives which will bear fruit in work that is in progress now.

The current matching algorithm is of very simple form. It is based on a similarity function that compares two jets (that is, strings of visual feature values for two locations, taken in two images or in the same image), and expresses the result as a "similarity value". This value is 1 if the jets are identical and is small when the jets differ strongly. We experimented with many different fuctional versions but so far we have always come back to the simple scalar product of normalized jets. In what follows we will refer to the layers $A^{(1)}$and $A^{(2)}$of Rosenblatt as the "image domain" and "model domain", respectively.

A stored object model has the form of a square array of points, each point carrying a jet taken from an image of the object. During model formation, the grid is positioned over the object. Elastic matching of a stored model to a new image proceeds in two steps. In step one the geometrical array of model points is positioned arbitrarily over the image. This establishes a tentative correspondende between model points and image points. Model points are now compared to the corresponding image points in terms of the similarity between model jets and image jets. Point similarities are added up to form a total similarity. Now, the model grid is moved rigidly over the image to optimize total similarity. Our experiments show that the total similarity is a fairly smooth function with a distinct and pronounced global optimum which corresponds to an acceptable position of the model grid over the image of the object.

In step two, individual match points are made to diffuse over the image plane in search of better local similarity, simultaneously optimizing, however, a "topology cost" along with the total model-image similarity. The topology term measures the deformation of the array of image points compared to the array of corresponding model points. This step of elastic deformation makes it possible to deal with elastic deformation of objects, caused, for instance, by rotation in depth. Details of the matching procedure are described in [11].

As part of the project we have extended the system to deal with invariance not only to object position but also to object size and orientation within the image plane. This is possible by estimating, in step one of the procedure, not only object position but also its size and orientation, scaling and rotating the grid of match points in the image domain, in addition to its translation. To compare image jets to model jets, it is now necessary to transform them along with the grid transformations. During this stage, it suffices according to our experience to compare only the levels of lowest resolution and to base initial comparisons only on a small number of model points. This corresponds to scaling, orienting and positioning the model in the image only in terms of a low pass filtered version of the model. An initial report of this work is published in [7], a full publication is in preparation.

2.3 Implementations of the System

During development of our system we have, in accord with the philosophy enunciated in the introduction, paid more attention to simplicity and generality of our system than to its parsimony in terms of computational requirements. The resulting system is rather expensive in terms of computation time. The most expensive step is the wavelet transform of the image. In our first version, the wavelet transform alone cost a good part of an hour on a work station. It is clear, however, that the wavelet transform can be computed with a high degree of parallelism and that correspondingly it can be performed in very little real time. We therefore had a high incentive to implement our system on parallel hardware.

As part of this project we acquired a system consisting of 23 transputers and developed OCCAMcode to implement the whole object recognition system on it (OCCAM is the parallel processing language for which the transputer has been developed [10, 13]). Initial development was hampered somewhat by the low level of support of the OCCAM development system for a large programming project. Once we had solved these teething problems, however, parallelizing our system turned out to be easy and very efficient. As the compute intensive parts of the recognition system consist of tasks working on independant pieces of data, we developed a general-purpose farming system. This has the additional features of automatic configuration and of allowing the broadcast of global read-only data (e.g., the preprocessed image as input to the matching step); for details, see [12]. Some additional, transputer-specific optimizations were introduced so that the system can make efficient use of all available transputers and exploits the full processing power of the transputers [12, 23]. Comparing an image to a database of some 90 objects with this system requires, on average, 19 seconds; this includes preprocessing.

The OCCAM system has the advantage of being efficient, but it is not a flexible basis for system development. We therefore have made an effort to create a platform for rapid system development. Our goal here is to create a system that allows rapid prototyping on a work station, making use of efficient interactive programming tools and graphics, and to download the most computing intensive routines into a parallel processing system. As part of the project we have achieved the following.

We first redesigned and implemented the OCCAM system in C++ (an object oriented C language) under the OpenWindows windowing system. This implementation makes full use of the inherent modularity of C++ by using objects to represent algorithmic features such as graphs, links and jets, as well as much of the graphical interface. Such a design allows only local modifications of a few objects to influence the whole structure and dynamics of the system. To illustrate this flexibility we replaced the previously used rectangular grid of jets by a more biologically plausible set of loci, the determination of which depends on hypothetized intrinsic saliencies of the faces [14]. This modification, otherwise costly in programing efforts, has been realized by the mere replacing of the graph object and its associated functions. In the same manner, as needs occurred, we added new display routines, allowing further monitoring of the state of the system, consequently gaining insights in its functioning and performance. However, the natural price to pay for modularity is computing efficiency. Taking again advantage of the modularity of this new system, we therefore undertook to create a set of efficient objects, the implementation of which could be run on the parallel transputer machine mentioned above.

We chose to use Trollius for this task, a Unix-based environment allowing the easy design and implementation of C code (rather than using OCCAM, or waiting for a C++ compiler) for the transputers. This environment is fully integrated into Unix and allows the easy mixing of Unix C (or C++) based code and transputer C code. We therefore implemented the routines performing the convolution of a 128 x 128 image with 40 kernels in C under Trollius and could reach performance levels similar to the OCCAMsystem. These routines have been designed in order to be ultimately used by the C++ based environment described above, as a new set of objects complementing those so far used and which were sequentially executed. However, both families of objects will coexist in the final version of the system. The sequential one will be used for prototyping whereas the parallel one will be used for actual experimentation, the user deciding at compile-time which familly is best suited for his/her needs. Ideally, this choice would be further allowed at run-time. An analogous implementation of the comparison routines is underway. Finally, ParaGraph, a set of graphical tools, has been installed and will be used to visualize the behavior of the parallel system (data bottlenecks, idle processors) and assess its overall performance.

3 Recognition Performance of the System

In order to assess the performance of the system, we collected three galleries of face images from 88 persons. One gallery is used to generate the database; the subjects were asked to look straight into the camera for this. A second gallery was taken with subjects looking approximately 20° to their right, while for the third gallery they were asked to modify their facial expression.

The comparison process described above yields a number for every pair of image and stored model. We thus need a mechanism to decide whether the model with the best match value is indeed the correct one, or whether a model of this person is not included in the database. For this, we developed two statictical criteria, described in detail in [11]. If the values for these criteria exceed a threshold, the model with the best match value is deemed recognized; otherwise, the system effectively says "I'm not sure."

The system's performance is given in the table. The thresholds of the criteria were adjusted such that for one gallery, no false positives resulted (columns 2 and 5). The system then identifies 88% and 84% of the images in a significant way (column 1), while at the same time it avoids wrong and significant recognitions (column 6) and false positives in the other gallery. In two (gallery 1) and three (gallery 2) cases, shown in column 4, the best match is not the correct model; conversely, in ≈97% of the cases, the system selects the correct model from the database (sum of columns 1 and 3).

3.1 Performance with Rotation and Scale Invariance

While the matching algorithm as described above will tolerate about 5–10% scale difference and about 10° rotation angle in the image plane between the image and a model, it would fail for larger variations of these global parameters. We thus implemented a system to estimate scale and orientation following a similar strategy as used for translation invariance. In place of a diffusion type optimization over different positions we adopted a hierarchical linear search for the optimal orientation and scale parameters.

The models are transformed according to the scale and orientation being searched and are then compared to the image. Since the affine transformation parameters depend on positioning, we used an interleaving scheme between the different global parameter estimation routines (positioning, scaling, rotation) and iterated from coarse to fine resolution. This procedure performed with over 90% recognition rate (85% if no false positives were allowed) on 90 test images showing objects at 70% and 50% the size of the database models (obtained through additional galleries of our subjects). A similar performance was obtained when the 70% gallery was additionally rotated by 30° in the image plane.

As result of the support in this project, the following papers have been published [12, 11, 7, 23, 14, 9, 20, 24, 18, 16, 25, 5, 8, 15, 19, ?, 4, 6, 1, 2, 3]

References

[1] M. A. Arbib and J. Buhmann. Neural networks. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 2, pages 1016–1060, New York, NY, USA, 1992. John Wiley.

[2] J. Buhmann and H. Kühnel. Complexity optimized vector quantization: A neural network approach. In J. Storer, editor, *Data Compression Conference*, pages 12–21, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.

[3] J. Buhmann and H. Kühnel. Unsupervised and supervised data clustering with competitive neural networks. In *IJCNN International Conference on Neural Networks, Baltimore*, pages IV–796 – IV–801. IEEE, 1992.

[4] J. Buhmann and H. Kühnel. Complexity optimized data clustering by competitive neural networks. *Neural Computation*, 5:75–88, 1993.

[5] J. Buhmann, J. Lange, C. v.d.Malsburg, J. C. Vorbrüggen, and R. P. Würtz. Object recognition in the dynamic link architecture — parallel implementation on a transputer network. In B. Kosko, editor, *Neural Networks: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall, New York, 1992.

[6] J. Buhmann and C. von der Malsburg. Sensory segmentation by neural oscillators. In *IJCNN International Conference on Neural Networks, Seattle*, pages II 603–607. IEEE, 1991.

[7] Joachim Buhmann, Martin Lades, and Christophvon der Malsburg. Size and distortion invariant object recognition by hierarchical graph matching. In *IJCNN International Conference on Neural Networks, SanDiego*, pages II 411–416. IEEE, 1990.

[8] S. Chandrashekhar, C. von der Malsburg, and R. Chellappa. Recursive tracking of image points using labelled graph matching. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Charlottesville, Virginia, October 1991.

[9] J.L. Gaudiot, C. von der Malsburg, and S. Shams. A data-flow implementation of a neurocomputer for pattern recognition applications. In *Proc. of the 1988 Aerospace Applications of Artificial Intelligence Conference*, Dayton, Ohio, 1988.

[10] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall International, Hemel Hempstead, 1989.

[11] Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, Jörg Lange, Christoph v.d. Malsburg, Rolf P. Würtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, March 1993.

[12] Martin Lades, Jan C. Vorbrüggen, and Rolf P. Würtz. Recognizing faces with a Transputer farm. In T.S. Durrani, W.A. Sandham, J.J. Soraghan, and S.M. Forbes, editors, *Applications of Transputers*, pages 148–153. IOS Press; Amsterdam, Oxford, Washington, Tokio, 1991.

[13] INMOS Limited. OCCAM 2 *Reference Manual.* Prentice Hall International, Hemel Hempstead, 1989.

[14] Manavendra Misra. *Implementation of Neural Networks on Parallel Architectures.* PhD thesis, Dept. of EE-Systems, University of Southern California, 1992.

[15] Manavendra Misra and V. K. Prasanna. Parallel computation of wavelet transforms. In *International Conference on Pattern Recognition*, Sept. 1992.

[16] C. v.d.Malsburg. Considerations for a visual architecture. In R. Eckmiller, editor, *Advanced Neural Computers*, pages 303–312, Amsterdam, 1990. North-Holland.

[17] C. von der Malsburg. Am i thinking assemblies? In G. Palm and A. Aertsen, editors, *Proceedings of the Trieste Meeting on Brain Theory.* Springer, Berlin, Heidelberg, October 1884 1986.

[18] C. von der Malsburg. A neural architecture for the representation of scenes. In J.L. McGaugh, N.M. Weinberger, and G. Lynch, editors, *Brain Organization and Memory: Cells, Systems and Circuits*, pages 356–372. Oxford University Press, New York, 1990.

[19] C. von der Malsburg and J. Buhmann. Sensory segmentation with coupled neural oscillators. *Biological Cybernetics*, 67:233–242, 1992.

[20] C. von der Malsburg, J. Buhmann, K. Flaton, and J. Lange. Vehicle identification in IR images, based on labeled graph matching. Project report, Hughes Aircraft Co., El Segundo, CA, 1988.

[21] Christoph von der Malsburg. The correlation theory of brain function. Technical report, Max-Planck-Institute for Biophysical Chemistry, Postfach 2841, Göttingen, FRG, 1981.

[22] Christoph von der Malsburg. Nervous structures with dynamical links. *Ber. Bunsenges. Phys. Chem.*, 89:703–710, 1985.

[23] J. C. Vorbrüggen. Parallelverarbeitung in Hardware: Optimierung numerischer Routinen auf dem T800. In *Transputer-Anwender-Treffen 1990*, Heidelberg Berlin New York, 1991. Springer Verlag.

[24] R. P. Würtz, J.C. Vorbrüggen, and C. v. d. Malsburg. A transputer system for the recognition of human faces by labeled graph matching. In R. Eckmiller, G. Hartmann, and G. Hauske, editors, *Parallel Processing in Neural Systems and Computers*, pages 37–41. North Holland, Amsterdam, 1990.

[25] Rolf P. Würtz, Jan C. Vorbrüggen, Christoph von der Malsburg, and Jörg Lange. A transputer-based neural object recognition system. In H. Burkhardt and J.C. Simon, editors, *From Pixels to Features II – Parallelism in Image Processing.* Elsevier, 1990.

gallery	criterion	case 1	2	3	4	5	6
gal. 1	κ_1	86	100	11	2	0	0
	κ_2	83	100	15	2	0	0
	κ	88	100	10	2	0	0
gal. 2	κ_1	79	100	17	3	0	0
	κ_2	80	100	16	3	0	0
	κ	84	100	13	3	0	0

Table 1: Results of comparing two galleries (gal. 1, head rotation by 20°; gal. 2, facial expressions) against the standard image database of the same persons. All entries are expressed as percentages. For details, see the text.